



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/787,226	02/27/2004	Ryan Mason	049051-0222	4844
31824	7590	03/06/2008	EXAMINER	
MCDERMOTT WILL & EMERY LLP			BELANI, KISHIN G	
18191 VON KARMAN AVE.				
SUITE 500			ART UNIT	PAPER NUMBER
IRVINE, CA 92612-7108			2143	
			MAIL DATE	DELIVERY MODE
			03/06/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/787,226	MASON ET AL.	
	Examiner	Art Unit	
	KISHIN G. BELANI	2143	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 14 December 2007.

2a) This action is **FINAL**. 2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-12 and 17-28 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) Claim(s) _____ is/are allowed.

6) Claim(s) 1-12 and 17-28 is/are rejected.

7) Claim(s) _____ is/are objected to.

8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All b) Some * c) None of:

1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. _____.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)

2) Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) Notice of Informal Patent Application

6) Other: _____.

DETAILED ACTION

This action is in response to Applicants' amendment filed on 12/14/2007.

Independent claims 1, 7, 17 and 21 have been amended by including additional limitations. **Claims 13-16 have been cancelled.** **Dependent claims 2-6, 8-12, 18-20 and 22-24 have been amended to correct minor informalities.** **New claims 25-28 have been added.** **Claims 1-12 and 17-28 are now pending** in the present application. Applicants' amendments to claims are shown in ***bold and italics*** in this office action, and the examiner's response to the amendments is shown in **bold**. **This Action is made FINAL.**

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 1, 4, 7, 10, 17,21 and 25-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Beadle et al. (U.S. Patent Publication # 7,039,709 B1)** in view of **Anne et al. (U.S. Patent Publication # 6,282,660 B1)**.

Consider **claim 1**, Beadle et al. show and disclose a user interface for managing a connection between a remote computing device and a local computing device (Fig. 4 that shows a GUI interface for connection management between a remote computing device (Web Browser 301 in Fig. 3) and a local computing device (Servers 303 (A-D) in Fig. 3); Fig. 5A that shows connection Settings GUI; column 2, lines 52-61 that disclose the same details), comprising:

a connection management window **operative to display** displaying at least a first connection icon (Fig. 5A that shows a connection management window; “Standard”

(should be DSL, see column 6, lines 64-66) connection icon 505A as a first connection icon),

the first connection icon representing a first connection between the remote computing device and a first local computing device, wherein in the connection management window a user can either select the first connection icon (DSL in Fig. 5A) or an active area (“Select Default Connection” area 501 in Fig. 5A) within the connection management window, wherein ~~if the user selection includes~~ **selecting** the first connection icon **allows a first** connection represented by the first connection icon **to** become modifiable to alter the first connection (using update options “Select Default Server” block 507, “Override Defaults” block 511, or “Automatic Routing” block 513, then clicking “Update Settings” button 515; column 7, lines 4-28 that disclose the same details), and

wherein ~~if the user selection includes~~ **selecting** the active area **allows** a new connection window **to** appear (a new connection window in Fig 5B, that permits selection of a more optimal route; column 11, lines 23-31 that disclose the same details) and, upon designating a new connection, **allows** a second connection icon **to be** displayed **within the connection management window** (Fig. 5A, “Satellite” block 505C), wherein the second connection icon represents a second connection different from the first connection, between the remote computing device and a second local computing device (column 11, lines 20-32 that disclose a second route connection; column 10, lines 6-15 that disclose two different network connections for the remote computing device).

However, Beadle et al. do not specifically disclose (only the bold italic features in the following text) a connection management window operative to display at least a first connection icon ***for a first application***; and upon designating a new connection, allows a second connection icon ***for a second application*** to be displayed within the connection management window; ***wherein the first application is different from the second application.***

In the same field of endeavor, Anne et al. disclose a connection management window operative to display at least a first connection icon ***for a first application***; and upon designating a new connection, allows a second connection icon ***for a second application*** to be displayed within the connection management window; ***wherein the first application is different from the second application*** (Fig. 3B, window 352 showing icon “Make New Connection”; column 5, lines 61-67 which disclose creating new connections using this icon; connection management window 352 displaying three icons created using “Make New Connection” icon, as shown in windows 353-355 and explained in column 5, lines 27-61; Fig. 4 that shows a first network application 401 using LAN driver 410 for connection and second applications 405 and 406 associated with dial services 404 using DSL/Cable modem 412 for connection, thereby disclosing a connection management window operative to display at least a first connection icon ***for a first application***; and upon designating a new connection, allows a second connection icon ***for a second application*** to be displayed within the connection management window; ***wherein the first application is different from the second application***).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a connection management window operative to display at least a first connection icon for a first application; and upon designating a new connection, allows a second connection icon for a second application to be displayed within the connection management window, as taught by Anne et al., in the user interface of Beadle et al., so as to provide a user a graphical interface to set up and manage network connections based on the applications, wherein the first application is different from the second application.

Consider **claim 4**, and **as it applies to claim 1 above**, Beadle et al., as modified by Anne et al., further disclose a user interface for managing a connection between a remote computing device and a local computing device, wherein the first connection icon and the second connection icon each includes a priority (in Beadle et al. reference, Fig. 5A, “Select Default Server” block 507, “Override Defaults” block 511, and “Update Settings” button 515 that enable a user to set priorities in selecting different connections and other options; Fig. 6A that lists some of the options 601 that can be assigned priority values to arrive at the relative ratings 605; column 8, lines 28-33 that disclose the same details).

Consider **claim 7**, Beadle et al. show and disclose a method for managing a connection between a local computing device and a remote computing device using a user interface (Fig. 4 that shows a GUI interface for connection management between a

remote computing device (Web Browser 301 in Fig. 3) and a local computing device (Servers 303 (A-D) in Fig. 3); Fig. 5A that shows connection Settings GUI; column 2, lines 52-61 that disclose the same details), comprising the steps of:

displaying a user interface (Fig. 4, that shows a default client browser connection 403 and a “Connections” icon 407 on the toolbar; column 6, lines 49-57 that disclose an improved user interface);

displaying at least a first connection icon on the user interface, the first connection icon representing a first connection between the remote computing device and a first local computing device (Fig. 5A that shows a connection management window; “Standard” (should be DSL, see column 6, lines 64-66) connection icon 505A as a first connection icon);

receiving a user selection of the first connection icon ***wherein the user selection of the first connection icon allows*** (DSL in Fig. 5A), ***a first*** connection represented by the first connection icon ***to*** become modifiable to alter the first connection (using update options “Select Default Server” block 507, “Override Defaults” block 511, or “Automatic Routing” block 513, then clicking “Update Settings” button 515; column 7, lines 4-28 that disclose the same details); and

receiving a user selection ***of*** an active area of the user interface, ***wherein the user selection of the active area allows*** (“Select Override Defaults” area 511 in Fig. 5A), a second connection icon ***to be*** displayed (Fig. 5A, “Satellite” block 505C), wherein the second connection icon represents a second connection different than the first connection (column 11, lines 20-32 that disclose a second route connection; column 10,

lines 6-15 that disclose two different network connections for the remote computing device).

However, Beadle et al. do not specifically disclose (only the bold italic features in the following text) displaying at least a first connection icon ***for a first application*** on the user interface; and displaying a second connection icon ***for a second application***.

In the same field of endeavor, Anne et al. disclose a connection management window displaying at least a first connection icon ***for a first application*** on the user interface; and displaying a second connection icon ***for a second application*** (Fig. 3B, **window 352 showing icon “Make New Connection”**; column 5, lines 61-67 which disclose creating new connections using this icon; connection management window 352 displaying three icons created using “Make New Connection” icon, as shown in windows 353-355 and explained in column 5, lines 27-61; Fig. 4 that shows a first network application 401 using LAN driver 410 for connection and second applications 405 and 406 associated with dial services 404 using DSL/Cable modem 412 for connection, thereby disclosing a connection management window displaying at least a first connection icon ***for a first application*** on the user interface; and displaying a second connection icon ***for a second application***).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a connection management window displaying at least a first connection icon for a first application on the user interface; and displaying a second connection icon for a second application, as taught by Anne et al.,

in the method of Beadle et al., so as to provide a user a graphical interface to set up and manage network connections based on the applications.

Consider **claim 10**, and **as it applies to claim 7 above**, Beadle et al., as modified by Anne et al., further disclose a method for managing a connection between a local computing device and a remote computing device using **a** user interface, wherein the first connection icon and the second connection icon each includes a priority (in beadle et al. reference, Fig. 5A, “Select Default Server” block 507, “Override Defaults” block 511, and “Update Settings” button 515 that enable a user to set priorities in selecting different connections and other options; Fig. 6A that lists some of the options 601 that can be assigned priority values to arrive at the relative ratings 605; column 8, lines 28-33 that disclose the same details).

Consider **claim 17**, Beadle et al. disclose computer-executable program code stored on a computer readable medium, said computer-executable program code for managing a connection between a local computing device and a remote computing device using a user interface (Claims 10-12 and 22-29 that disclose and claim the same details; Fig. 4 that shows a GUI interface for connection management between a remote computing device (Web Browser 301 in Fig. 3) and a local computing device (Servers 303 (A-D) in Fig. 3); Fig. 5A that shows connection Settings GUI; column 2, lines 52-61 that disclose the same details), the computer-executable program code comprising:

code for displaying a user interface (Fig. 4, that shows a default client browser connection 403 and a “Connections” icon 407 on the toolbar; column 6, lines 49-57 that disclose an improved user interface);

code for displaying at least a first connection icon on the user interface, the first connection icon representing a first connection between the remote computing device and a first local computing device (Fig. 5A that shows a connection management window; “Standard” (should be DSL, see column 6, lines 64-66) connection icon 505A as a first connection icon);

code for receiving a user selection of the first connection icon (DSL in Fig. 5A), **wherein the user selection of the first connection icon allows a first** connection represented by the first connection icon to become modifiable to alter the first connection (using update options “Select Default Server” block 507, “Override Defaults” block 511, or “Automatic Routing” block 513, then clicking “Update Settings” button 515; column 7, lines 4-28 that disclose the same details); and

code for receiving a user selection of an active area of the user interface (“Select Override Defaults” area 511 in Fig. 5A), **wherein the user selection of the active area allows** a second connection icon to be displayed (Fig. 5A, “Satellite” block 505C), wherein the second connection icon represents a second connection different than the first connection (column 11, lines 20-32 that disclose a second route connection; column 10, lines 6-15 that disclose two different network connections for the remote computing device).

However, Beadle et al. do not specifically disclose (only the bold italic features in the following text) code for displaying at least a first connection icon ***for a first application*** on the user interface; and code for displaying a second connection icon ***for a second application***.

In the same field of endeavor, Anne et al. disclose code for a connection management window displaying at least a first connection icon ***for a first application*** on the user interface; and code for displaying a second connection icon ***for a second application*** (claim 14; Fig. 3B, window 352 showing icon “Make New Connection”; column 5, lines 61-67 which disclose creating new connections using this icon; connection management window 352 displaying three icons created using “Make New Connection” icon, as shown in windows 353-355 and explained in column 5, lines 27-61; Fig. 4 that shows a first network application 401 using LAN driver 410 for connection and second applications 405 and 406 associated with dial services 404 using DSL/Cable modem 412 for connection, thereby disclosing code for a connection management window displaying at least a first connection icon ***for a first application*** on the user interface; and displaying a second connection icon ***for a second application***).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide code for a connection management window displaying at least a first connection icon for a first application on the user interface; and code for displaying a second connection icon for a second application, as taught by Anne et al., in the program code (claims 10-12) of Beadle et al., so as to provide a user

a graphical interface to set up and manage network connections based on the applications.

Consider **claim 21**, Beadle et al. show and disclose a programmed computer apparatus for managing a connection between a local computing device and a remote computing device using a user interface (Fig. 4 that shows a GUI interface for connection management between a remote computing device (Web Browser 301 in Fig. 3) and a local computing device (Servers 303 (A-D) in Fig. 3); Fig. 5A that shows connection Settings GUI; column 2, lines 52-61 that disclose the same details), said programmed computer apparatus comprising:

means for displaying a user interface (Fig. 4, that shows a default client browser connection 403 and a “Connections” icon 407 on the toolbar; column 6, lines 49-57 that disclose an improved user interface);

means for displaying at least a first connection icon on the user interface, the first connection icon representing a first connection between the remote computing device and a first local computing device (Fig. 5A that shows a connection management window; “Standard” (should be DSL, see column 6, lines 64-66) connection icon 505A as a first connection icon);

means for receiving a user selection of the first connection icon (DSL in Fig. 5A), ***wherein the user selection of the first connection icon allows a first*** connection represented by the first connection icon to become modifiable to alter the first connection (using update options “Select Default Server” block 507, “Override Defaults”

block 511, or “Automatic Routing” block 513, then clicking “Update Settings” button 515; column 7, lines 4-28 that disclose the same details); and

means for receiving a user selection of an active area of the user interface (“Select Override Defaults” area 511 in Fig. 5A), ***wherein the user selection of the active area allows*** a second connection icon to be displayed (Fig. 5A, “Satellite” block 505C), wherein the second connection icon represents a second connection different than the first connection (column 11, lines 20-32 that disclose a second route connection; column 10, lines 6-15 that disclose two different network connections for the remote computing device).

However, Beadle et al. do not specifically disclose (only the bold italic features in the following text) means for displaying at least a first connection icon ***for a first application*** on the user interface; and for displaying a second connection icon ***for a second application.***

In the same field of endeavor, Anne et al. disclose code for a connection management window displaying at least a first connection icon ***for a first application*** on the user interface; and code for displaying a second connection icon ***for a second application*** (claim 14; Fig. 3B, window 352 showing icon “Make New Connection”; column 5, lines 61-67 which disclose creating new connections using this icon; connection management window 352 displaying three icons created using “Make New Connection” icon, as shown in windows 353-355 and explained in column 5, lines 27-61; Fig. 4 that shows a first network application 401 using LAN driver 410 for connection and second applications 405 and 406 associated with dial services

404 using DSL/Cable modem 412 for connection, thereby disclosing code for a connection management window displaying at least a first connection icon ***for a first application*** on the user interface; and displaying a second connection icon ***for a second application***).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide code for a connection management window displaying at least a first connection icon for a first application on the user interface; and code for displaying a second connection icon for a second application, as taught by Anne et al., in the program code (claims 10-12) of Beadle et al., so as to provide a user a graphical interface to set up and manage network connections based on the applications.

Consider **claim 25**, and **as it applies to claim 1 above**, Beadle et al., as modified by Anne et al., further disclose the claimed user interface, wherein the remote computing device is a thin client (in Beadle et al. reference, column 1, lines 32-34 which disclose that clients can be “dumber” systems (thin clients) adapted for limited use with a network); wherein the user interface is to be displayed at the thin client (column 2, lines 54-57 that disclose a graphical user interface for receiving user selection at the remote thin client, and a connection utility for connecting the client with a selected local server); wherein the active area is an empty area (in Beadle et al. reference, Fig. 5A, desktop area 507, 511 or 513 that does not include the first connection areas 505A-D (this interpretation of empty area based on the applicants’ definition of empty area in

paragraph 0075 of the application stating that empty area is where desktop is not displaying first connection)), and wherein selecting the first connection icon allows the user to edit or delete the first connection (in Beadle et al. reference, Fig. 5A, “Select Default Server” block 507, “Override Defaults” block 511, and “Update Settings” button 515 that enable a user to edit the first connection by selecting an alternate server (as shown in Fig. 5B); column 6, lines 58-67 through column 7, lines 1-30 which disclose that after selecting a first connection icon (505A-D), the user has option to edit the connection by clicking on the “Update Settings” button).

Consider **claim 26**, and **as it applies to claim 7 above**, Beadle et al., as modified by Anne et al., further disclose the claimed method, wherein the remote computing device is a thin client (in Beadle et al. reference, column 1, lines 32-34 which disclose that clients can be “dumber” systems (thin clients) adapted for limited use with a network); wherein the user interface is to be displayed at the thin client (column 2, lines 54-57 that disclose a graphical user interface for receiving user selection at the remote thin client, and a connection utility for connecting the client with a selected local server); wherein the active area is an empty area (in Beadle et al. reference, Fig. 5A, desktop area 507, 511 or 513 that does not include the first connection areas 505A-D (this interpretation of empty area based on the applicants’ definition of empty area in paragraph 0075 of the application stating that empty area is where desktop is not displaying first connection));

wherein selecting the first connection icon allows the user to edit or delete the first connection (in Beadle et al. reference, Fig. 5A, “Select Default Server” block 507, “Override Defaults” block 511, and “Update Settings” button 515 that enable a user to edit the first connection by selecting an alternate server (as shown in Fig. 5B); column 6, lines 58-67 through column 7, lines 1-30 which disclose that after selecting a first connection icon (505A-D), the user has option to edit the connection by clicking on the “Update Settings” button);

wherein the second connection is between the thin client and a second local computing device (in Beadle et al. reference, Fig. 10 that shows a second connection using modem transmission; column 10, lines 6-24 which disclose a first connection via satellite to a DirectPC application and a second modem connection to a server for a financial application); and

wherein the first application is different from the second application (in Anne et al. reference, Fig. 4 that shows a first network application 401 using LAN driver 410 for connection and second applications 405 and 406 associated with dial services 404 using DSL/Cable modem 412 for connection, thereby disclosing that the first application is different from the second application).

Consider **claim 27**, and **as it applies to claim 17 above**, Beadle et al., as modified by Anne et al., further disclose the claimed computer-executable program code, wherein the remote computing device is a thin client (in Beadle et al. reference,

claims 10-12; column 1, lines 32-34 which disclose that clients can be “dumber” systems (thin clients) adapted for limited use with a network); wherein the user interface is to be displayed at the thin client (column 2, lines 54-57 that disclose a graphical user interface for receiving user selection at the remote thin client, and a connection utility for connecting the client with a selected local server); wherein the active area is an empty area (in Beadle et al. reference, Fig. 5A, desktop area 507, 511 or 513 that does not include the first connection areas 505A-D (this interpretation of empty area based on the applicants’ definition of empty area in paragraph 0075 of the application stating that empty area is where desktop is not displaying first connection)); wherein the user selection of the first connection icon allows a user at the thin client to edit or delete the first connection (in Beadle et al. reference, Fig. 5A, “Select Default Server” block 507, “Override Defaults” block 511, and “Update Settings” button 515 that enable a user to edit the first connection by selecting an alternate server (as shown in Fig. 5B); column 6, lines 58-67 through column 7, lines 1-30 which disclose that after selecting a first connection icon (505A-D), the user has option to edit the connection by clicking on the “Update Settings” button); wherein the second connection is between the thin client and a second local computing device (in Beadle et al. reference, Fig. 10 that shows a second connection using modem transmission; column 10, lines 6-24 which disclose a first connection via satellite to a DirectPC application and a second modem connection to a server for a financial application); and

wherein the first application is different from the second application (in Anne et al. reference, Fig. 4 that shows a first network application 401 using LAN driver 410 for connection and second applications 405 and 406 associated with dial services 404 using DSL/Cable modem 412 for connection, thereby disclosing that the first application is different from the second application).

Consider **claim 28**, and **as it applies to claim 21 above**, Beadle et al., as modified by Anne et al., further disclose the claimed programmed computer apparatus, wherein the remote computing device is a thin client (in Beadle et al. reference, claims 10-12; column 1, lines 32-34 which disclose that clients can be “dumber” systems (thin clients) adapted for limited use with a network); wherein the user interface is to be displayed at the thin client (column 2, lines 54-57 that disclose a graphical user interface for receiving user selection at the remote thin client, and a connection utility for connecting the client with a selected local server); wherein the active area is an empty area (in Beadle et al. reference, Fig. 5A, desktop area 507, 511 or 513 that does not include the first connection areas 505A-D (this interpretation of empty area based on the applicants’ definition of empty area in paragraph 0075 of the application stating that empty area is where desktop is not displaying first connection)); wherein the user selection of the first connection icon allows a user at the thin client to edit or delete the first connection (in Beadle et al. reference, Fig. 5A, “Select Default Server” block 507, “Override Defaults” block 511, and “Update Settings” button 515 that

enable a user to edit the first connection by selecting an alternate server (as shown in Fig. 5B); column 6, lines 58-67 through column 7, lines 1-30 which disclose that after selecting a first connection icon (505A-D), the user has option to edit the connection by clicking on the "Update Settings" button);

wherein the second connection is between the thin client and a second local computing device (in Beadle et al. reference, Fig. 10 that shows a second connection using modem transmission; column 10, lines 6-24 which disclose a first connection via satellite to a DirectPC application and a second modem connection to a server for a financial application); and

wherein the first application is different from the second application (in Anne et al. reference, Fig. 4 that shows a first network application 401 using LAN driver 410 for connection and second applications 405 and 406 associated with dial services 404 using DSL/Cable modem 412 for connection, thereby disclosing that the first application is different from the second application).

Claims 2, 8, 18 and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Beadle et al. (U.S. Patent Publication # 7,039,709 B1)** in view of **Anne et al. (U.S. Patent Publication # 6,282,660 B1)** and further in view of **Loisey et al. (U.S. Patent Publication # 6,999,912 B2)**.

Consider **claim 2**, and **as it applies to claim 1 above**, Beadle et al., as modified by Anne et al., show and disclose a user interface for managing a connection between a

remote computing device and a local computing device, except wherein the connection management window ***is operative to display*** at least a first application icon, wherein the first application icon represents an application available for execution on the first local computing device.

In the same field of endeavor, Loisey et al. do show and disclose an improved user interface, wherein the connection management window displays at least a first application icon, wherein the first application icon represents an application available for execution on the first local computing device (Fig. 3, a first application icon 320 along with a connection management icon (part of Computer Settings icon 310); column 7, lines 59-67 that disclose the same details).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to display at least a first application icon in the connection management window, wherein the first application icon represents an application available for execution on the first local computing device, as taught by Loisey et al., in the user interface of Beadle et al., as modified by Anne et al., so as to provide particular needs and desires of the user.

Consider **claim 8**, and **as it applies to claim 7 above**, Beadle et al., as modified by Anne et al., show and disclose a method for managing a connection between a local computing device and a remote computing device using a user interface, except further comprising the step of displaying at least a first application icon, wherein the first

application icon represents an application available for execution on the first local computing device.

In the same field of endeavor, Loisey et al. do show and disclose a method for managing a connection between a local computing device and a remote computing device using an improved user interface, further comprising the step of displaying at least a first application icon, wherein the first application icon represents an application available for execution on the first local computing device (Fig. 3, a first application icon 320 along with a connection management icon (part of Computer Settings icon 310); column 7, lines 59-67 that disclose the same details).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to display at least a first application icon, wherein the first application icon represents an application available for execution on the first local computing device, as taught by Loisey et al., in the method of Beadle et al., as modified by Anne et al., so as to provide particular needs and desires of the user.

Consider **claim 18**, and **as it applies to claim 17 above**, Beadle et al., as modified by Anne et al., disclose a computer-executable program code, except said program code comprising code for displaying at least a first application icon, wherein the first application icon represents an application available for execution on the first local computing device.

In the same field of endeavor, Loisey et al. disclose a computer-executable program code for displaying at least a first application icon, wherein the first application

icon represents an application available for execution on the first local computing device (Claims 1-5; Fig. 3, a first application icon 320 along with a connection management icon (part of Computer Settings icon 310); column 7, lines 59-67 that disclose the same details).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide code to display at least a first application icon in the connection management window, wherein the first application icon represents an application available for execution on the first local computing device, as taught by Loisey et al., in the computer-readable storage medium with stored program code of Beadle et al., as modified by Anne et al., so as to provide particular needs and desires of the user.

Consider **claim 22**, and **as it applies to claim 21 above**, Beadle et al., as modified by Anne et al., show and disclose a programmed computer apparatus for managing a connection between a local computing device and a remote computing device using a user interface, except wherein said programmed computer apparatus comprising means for displaying at least a first application icon, wherein the first application icon represents an application available for execution on the first local computing device.

In the same field of endeavor, Loisey et al. do show and disclose a programmed computer apparatus for managing a connection between a remote computing device and a local computing device, wherein said programmed computer apparatus

comprising means for displaying at least a first application icon, wherein the first application icon represents an application available for execution on the first local computing device (Fig. 3, a first application icon 320 along with a connection management icon (part of Computer Settings icon 310); column 7, lines 59-67 that disclose the same details).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a programmed computer apparatus comprising means for displaying at least a first application icon, wherein the first application icon represents an application available for execution on the first local computing device, as taught by Loisey et al., in the programmed computer apparatus of Beadle et al., as modified by Anne et al., so as to provide particular needs and desires of the user.

Claims 3, 9, 19 and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Beadle et al. (U.S. Patent Publication # 7,039,709 B1)** in view of **Anne et al. (U.S. Patent Publication # 6,282,660 B1)** and further in view of **Perholtz et al. (U.S. Patent Application Publication # 2002/0091850 A1)**.

Consider **claim 3**, and **as it applies to claim 1 above**, Beadle et al., as modified by Anne et al., show and disclose a user interface for managing a connection between a remote computing device and a local computing device, except further comprising a keystroke management window, wherein the keystroke management window is user

modifiable to accept a local keystroke management setting, wherein if the local keystroke management setting is enabled, a keystroke is processed at the remote computing device, and wherein if the local keystroke management setting is disabled, the keystroke is processed at the first local computing device.

In the same field of endeavor, Perholtz et al. disclose a user interface, wherein the keystroke management is user modifiable to accept a local keystroke management setting, wherein if the local keystroke management setting is enabled, a keystroke is processed at the remote computing device, and wherein if the local keystroke management setting is disabled, the keystroke is processed at the first local computing device (Flowchart of Fig. 7G, decision block 759 that checks for use of “Hot Keys”; paragraph 0288, lines 1-16 that disclose the use of “Hot Keys” for redirecting remote client’s input keystrokes/mouse data to the local server and means to return back to the remote client’s normal mode of operation by tapping the left shift key three times within 2 seconds; although no window is shown for selecting an option to make hot key sequences effective either at a local computing device or at a remote computing device, the paragraph does mention selecting from a menu, either Remote PC mode or Host mode. Based on the user selection, the hot keys are either applicable at the remote computing device, or at the local computing device. Therefore, the examiner has taken the official notice that the use of keystrokes achieves the same purpose as the mouse clicks on a GUI interface, as is evident when copying a paragraph from one document and pasting it into another document. One may use Ctrl-c keyboard keys to copy a

selected paragraph or use a pulldown menu (GUI) or a toolbar icon to copy the paragraph).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a user interface, wherein the keystroke management is user modifiable to accept a local keystroke management setting, wherein if the local keystroke management setting is enabled, a keystroke is processed at the remote computing device, and wherein if the local keystroke management setting is disabled, the keystroke is processed at the first local computing device, as taught by Perholtz et al., in the user interface of Beadle et al., as modified by Anne et al., so as to provide a user ability to use hot keys to execute applications at the local computing device as well as at the remote computing device, and be able to easily switch between them.

Consider **claim 9**, and **as it applies to claim 7 above**, Beadle et al., as modified by Anne et al., show and disclose a method for managing a connection between a local computing device and a remote computing device using a user interface, except further comprising the step of displaying a keystroke management window, wherein the keystroke management window is user modifiable to accept a local keystroke management setting, wherein if the local keystroke management setting is enabled, a keystroke is processed at the remote computing device, and wherein if the local keystroke management setting is disabled, the keystroke is processed at the first local computing device.

In the same field of endeavor, Perholtz et al. disclose a method, further comprising the step of displaying a keystroke management window, wherein the keystroke management window is user modifiable to accept a local keystroke management setting, wherein if the local keystroke management setting is enabled, a keystroke is processed at the remote computing device, and wherein if the local keystroke management setting is disabled, the keystroke is processed at the first local computing device (Flowchart of Fig. 7G, decision block 759 that checks for use of “Hot Keys”; paragraph 0288, lines 1-16 that disclose the use of “Hot Keys” for redirecting remote client’s input keystrokes/mouse data to the local server and means to return back to the remote client’s normal mode of operation by tapping the left shift key three times within 2 seconds; although no window is shown for selecting an option to make hot key sequences effective either at a local computing device or at a remote computing device, the paragraph does mention selecting from a menu, either Remote PC mode or Host mode. Based on the user selection, the hot keys are either applicable at the remote computing device, or at the local computing device. Therefore, the examiner has taken the official notice that the use of keystrokes achieves the same purpose as the mouse clicks on a GUI interface, as is evident when copying a paragraph from one document and pasting it into another document. One may use Ctrl-c keyboard keys to copy a selected paragraph or use a pulldown menu (GUI) or a toolbar icon to copy the paragraph).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a method, further comprising the step of

displaying a keystroke management window, wherein the keystroke management window is user modifiable to accept a local keystroke management setting, wherein if the local keystroke management setting is enabled, a keystroke is processed at the remote computing device, and wherein if the local keystroke management setting is disabled, the keystroke is processed at the first local computing device, as taught by Perholtz et al., in the method of Beadle et al., as modified by Anne et al., so as to provide a user ability to use hot keys to execute applications at the local computing device as well as at the remote computing device, and be able to easily switch between them.

Consider **claim 19**, and **as it applies to claim 17 above**, Beadle et al., as modified by Anne et al., disclose a computer-executable program code, except said program code comprising code for displaying a keystroke management window, wherein the keystroke management window is user modifiable to accept a local keystroke management setting, wherein if the local keystroke management setting is enabled, a keystroke is processed at the remote computing device, and wherein if the local keystroke management setting is disabled, the keystroke is processed at the first local computing device.

In the same field of endeavor, Perholtz et al. disclose a computer-readable storage medium with stored program code for managing a connection between a local computing device and a remote computing device using an improved user interface, said program comprising code for displaying a keystroke management window, wherein

the keystroke management window is user modifiable to accept a local keystroke management setting, wherein if the local keystroke management setting is enabled, a keystroke is processed at the remote computing device, and wherein if the local keystroke management setting is disabled, the keystroke is processed at the first local computing device (Claims 1-5; Flowchart of Fig. 7G, decision block 759 that checks for use of “Hot Keys”; paragraph 0288, lines 1-16 that disclose the use of “Hot Keys” for redirecting remote client’s input keystrokes/mouse data to the local server and means to return back to the remote client’s normal mode of operation by tapping the left shift key three times within 2 seconds; although no window is shown for selecting an option to make hot key sequences effective either at a local computing device or at a remote computing device, the paragraph does mention selecting from a menu, either Remote PC mode or Host mode. Based on the user selection, the hot keys are either applicable at the remote computing device, or at the local computing device. Therefore, the examiner has taken the official notice that the use of keystrokes achieves the same purpose as the mouse clicks on a GUI interface, as is evident when copying a paragraph from one document and pasting it into another document. One may use Ctrl-c keyboard keys to copy a selected paragraph or use a pulldown menu (GUI) or a toolbar icon to copy the paragraph).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a computer-readable storage medium with stored program code for managing a connection between a local computing device and a remote computing device using an improved user interface, said program comprising

code for displaying a keystroke management window, wherein the keystroke management window is user modifiable to accept a local keystroke management setting, wherein if the local keystroke management setting is enabled, a keystroke is processed at the remote computing device, and wherein if the local keystroke management setting is disabled, the keystroke is processed at the first local computing device, as taught by Perholtz et al., in the program code of Beadle et al., as modified by Anne et al., so as to provide a user ability to use hot keys to execute applications at the local computing device as well as at the remote computing device, and be able to easily switch between them.

Consider **claim 23**, and **as it applies to claim 21 above**, Beadle et al., as modified by Anne et al., show and disclose a programmed computer apparatus for managing a connection between a remote computing device and a local computing device using a user interface, except comprising means for displaying a keystroke management window, wherein the keystroke management window is user modifiable to accept a local keystroke management setting, wherein if the local keystroke management setting is enabled, a keystroke is processed at the remote computing device, and wherein if the local keystroke management setting is disabled, the keystroke is processed at the first local computing device.

In the same field of endeavor, Perholtz et al. disclose a programmed computer apparatus, wherein the keystroke management is user modifiable to accept a local keystroke management setting, wherein if the local keystroke management setting is

enabled, a keystroke is processed at the remote computing device, and wherein if the local keystroke management setting is disabled, the keystroke is processed at the first local computing device (Flowchart of Fig. 7G, decision block 759 that checks for use of “Hot Keys”; paragraph 0288, lines 1-16 that disclose the use of “Hot Keys” for redirecting remote client’s input keystrokes/mouse data to the local server and means to return back to the remote client’s normal mode of operation by tapping the left shift key three times within 2 seconds; although no window is shown for selecting an option to make hot key sequences effective either at a local computing device or at a remote computing device, the paragraph does mention selecting from a menu, either Remote PC mode or Host mode. Based on the user selection, the hot keys are either applicable at the remote computing device, or at the local computing device. Therefore, the examiner has taken the official notice that the use of keystrokes achieves the same purpose as the mouse clicks on a GUI interface, as is evident when copying a paragraph from one document and pasting it into another document. One may use Ctrl-c keyboard keys to copy a selected paragraph or use a pull-down menu (GUI) or a toolbar icon to copy the paragraph).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a programmed computer apparatus, wherein the keystroke management is user modifiable to accept a local keystroke management setting, wherein if the local keystroke management setting is enabled, a keystroke is processed at the remote computing device, and wherein if the local keystroke management setting is disabled, the keystroke is processed at the first local

computing device, as taught by Perholtz et al., in the programmed computer apparatus of Beadle et al., as modified by Anne et al., so as to provide a user ability to use hot keys to execute applications at the local computing device as well as at the remote computing device, and be able to easily switch between them.

Claims 5 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Beadle et al. (U.S. Patent Publication # 7,039,709 B1)** in view of **Anne et al. (U.S. Patent Publication # 6,282,660 B1)** and further in view of **Lele (U.S. Patent Publication # 7,181,524 B1)**.

Consider **claim 5**, and **as it applies to claim 1 above**, Beadle et al., as modified by Anne et al., show and disclose a user interface for managing a connection between a remote computing device and a local computing device, except, wherein the priority is a failover order.

In the same field of endeavor, Lele discloses a user interface, wherein the priority is a failover order (column 1, lines 21-27 that disclose a plurality of servers connected in a server cluster to provide failover redundancy; Fig. 1, Rules block 154 and Selection Algorithm block 155 that specify server selection criteria; thereby disclosing server failover order that a user may specify as a priority option in the connection management).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a user interface for managing a connection

between a remote computing device and a local computing device, wherein the priority is a failover order, as taught by Lele, in the user interface of Beadle et al., as modified by Anne et al., so as to provide an alternate connection path to a server, in case the selected server fails.

Consider **claim 11**, and **as it applies to claim 7 above**, Beadle et al., as modified by Anne et al., show and disclose a method for managing a connection between a local computing device and a remote computing device using a user interface, except wherein the priority is a failover order.

In the same field of endeavor, Lele discloses a method, wherein the priority is a failover order (column 1, lines 21-27 that disclose a plurality of servers connected in a server cluster to provide failover redundancy; Fig. 1, Rules block 154 and Selection Algorithm block 155 that specify server selection criteria; thereby disclosing server failover order that a user may specify as a priority option in the connection management).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a method for managing a connection between a local computing device and a remote computing device using an improved user interface, wherein the priority is a failover order, as taught by Lele, in the method of Beadle et al., as modified by Anne et al., so as to provide an alternate connection path to a server, in case the selected server fails.

Claims 6, 12, 20 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Beadle et al. (U.S. Patent Publication # 7,039,709 B1)** in view of **Anne et al. (U.S. Patent Publication # 6,282,660 B1)** and further in view of **Ritchy et al. (U.S. Patent Application Publication # 2004/0183831 A1)**.

Consider **claim 6**, and **as it applies to claim 1 above**, Beadle et al., as modified by Anne et al., show and disclose a user interface for managing a connection between a remote computing device and a local computing device, except further comprising a desktop shell window, wherein the desktop shell window is modifiable **at run time by the user at the remote computing device** to accept a desktop shell setting, wherein if the desktop shell setting is disabled, an alternate user interface is selected and the user interface is disabled.

In the same field of endeavor, Ritchy et al. disclose a desktop window, wherein the desktop shell window is modifiable **at run time by the user at the remote computing device** to accept a desktop shell setting, wherein if the desktop shell setting is disabled, an alternate user interface is selected and the improved user interface is disabled (Fig. 9 that shows a default desktop window and a pull-down to select alternate desktop shell if the user so desires; paragraph 0049, lines 9-11 which disclose that different shells for the desktop are selectable in the Property Editor window, and portal administrators and end users can also change a desktop's shell, thereby disclosing that the desktop shell window is modifiable at run time by the user at the remote computing device to accept a desktop shell setting, selecting an alternate user interface, if the

desktop shell setting is disabled, and disabling the improved user interface, if the desktop shell setting is disabled).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a desktop shell window, wherein the desktop shell window is modifiable at run time by the user at the remote computing device to accept a desktop shell setting, wherein if the desktop shell setting is disabled, an alternate user interface is selected and the improved user interface is disabled, as taught by Ritchy et al., in the user interface of Beadle et al., as modified by Anne et al., so as to provide multiple operating systems environments for the user to choose from, based on user's preferences, on the same desktop.

Consider **claim 12**, and **as it applies to claim 7 above**, Beadle et al., as modified by Anne et al., show and disclose a method for an improved user interface for managing a connection between a local computing device and a remote computing device using a user interface, except further comprising the steps of displaying a desktop shell window, wherein the desktop shell window is modifiable **at run-time by a user at the remote computing device** to accept a desktop shell setting; selecting an alternate user interface, if the desktop shell setting is disabled; and disabling the improved user interface, if the desktop shell setting is disabled.

In the same field of endeavor, Ritchy et al. disclose a method for displaying a desktop shell window, wherein the desktop shell window is modifiable **at run time by the user at the remote computing device** to accept a desktop shell setting, selecting

an alternate user interface, if the desktop shell setting is disabled, and disabling the improved user interface, if the desktop shell setting is disabled (Fig. 9 that shows a default desktop window with a user interface (pull-down) to select an alternate desktop shell if the user so desires; paragraph 0049, lines 9-11 which disclose that different shells for the desktop are selectable in the Property Editor window, and portal administrators and end users can also change a desktop's shell, thereby disclosing that the desktop shell window is modifiable at run time by the user at the remote computing device to accept a desktop shell setting, selecting an alternate user interface, if the desktop shell setting is disabled, and disabling the improved user interface, if the desktop shell setting is disabled).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a method comprising the steps of displaying a desktop shell window, wherein the desktop shell window is modifiable at run time by the user at the remote computing device to accept a desktop shell setting, selecting an alternate user interface, if the desktop shell setting is disabled, and disabling the improved user interface, if the desktop shell setting is disabled, as taught by Ritchy et al., in the method of Beadle et al., as modified by Anne et al., so as to provide multiple operating systems environments for the user to choose from, based on user's preferences, on the same desktop.

Consider **claim 20**, and **as it applies to claim 17 above**, Beadle et al., as modified by Anne et al., disclose a computer-executable program code, except said

program code comprising code for displaying a desktop shell window, wherein the desktop shell window is modifiable ***at run-time by a user at the remote computing device*** to accept a desktop shell setting; code for selecting an alternate user interface, if the desktop shell setting is disabled; and code for disabling the user interface, if the desktop shell setting is disabled.

In the same field of endeavor, Ritchy et al. disclose a computer-readable storage medium with stored program code for managing a connection between a local computing device and a remote computing device, said program comprising codes for permitting the computer to perform a step for displaying a desktop shell window, wherein the desktop shell window is modifiable ***at run-time by a user at the remote computing device*** to accept a desktop shell setting; a selecting step for selecting an alternate user interface, if the desktop shell setting is disabled; a disabling step for disabling the improved user interface, if the desktop shell setting is disabled (Claims 20-38, 60-80, and 101-120; that shows a default desktop window with a user interface (pull-down) to select an alternate desktop shell if the user so desires; paragraph 0049, lines 9-11 which disclose that different shells for the desktop are selectable in the Property Editor window, and portal administrators and end users can also change a desktop's shell, thereby disclosing that the desktop shell window is modifiable at run-time by a user at the remote computing device to accept a desktop shell setting, selecting an alternate user interface, if the desktop shell setting is disabled, and disabling the improved user interface, if the desktop shell setting is disabled).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a computer-readable storage medium with stored program code for managing a connection between a local computing device and a remote computing device, said program comprising codes for permitting the computer to perform a step for displaying a desktop shell window, wherein the desktop shell window is modifiable at run-time by a user at the remote computing device to accept a desktop shell setting; a selecting step for selecting an alternate user interface, if the desktop shell setting is disabled; a disabling step for disabling the improved user interface, if the desktop shell setting is disabled, as taught by Ritchy et al., in the program code of Beadle et al., as modified by Anne et al., so as to provide a user ability to select any one of the many available desktop shells that is most suited to the user.

Consider **claim 24**, and **as it applies to claim 21 above**, Beadle et al., as modified by Anne et al., show and disclose a programmed computer apparatus for managing a connection between a local computing device and a remote computing device using a user interface, except comprising means for displaying a desktop shell window, wherein the desktop shell window is modifiable ***at run time by a user at the remote computing device***; means for selecting an alternate user interface, if the desktop shell setting is disabled; and means for disabling the user interface, if the desktop shell setting is disabled.

In the same field of endeavor, Ritchy et al. disclose a a programmed computer apparatus with a desktop shell window, wherein the desktop shell window is modifiable

at run time by a user at the remote computing device; to accept a desktop shell setting, means for selecting an alternate user interface, if the desktop shell setting is disabled, and means for disabling the improved user interface, if the desktop shell setting is disabled (Fig. 9 that shows a default desktop window and a pull-down to select alternate desktop shell if the user so desires; paragraph 0049, lines 9-11 which disclose that different shells for the desktop are selectable in the Property Editor window, and portal administrators and end users can also change a desktop's shell, thereby disclosing that the desktop shell window is modifiable at run time by a user at the remote computing device; to accept a desktop shell setting, means for selecting an alternate user interface, if the desktop shell setting is disabled, and means for disabling the improved user interface, if the desktop shell setting is disabled).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a desktop shell window, wherein the desktop shell window is modifiable at run time by a user at the remote computing device; to accept a desktop shell setting, means for selecting an alternate user interface, if the desktop shell setting is disabled, and means for disabling the improved user interface, if the desktop shell setting is disabled, as taught by Ritchy et al., in the programmed computer apparatus of Beadle et al., as modified by Anne et al., so as to provide multiple operating systems environments for the user to choose from, based on user's preferences, on the same desktop.

Response to Arguments

Applicants' arguments with respect to **claims 1-12 and 17-24** have been considered but are moot in view of the new ground(s) for rejection. However, in order to further clarify the grounds for rejecting these claims, the examiner has provided the following response to the arguments raised by the applicants:

The examiner respectfully disagrees with applicants' arguments as the applied references provide adequate support and clarification for claims rejection. Therefore, the examiner's **rejection of 09/11/2007 for claims 1-12 and 17-24** is maintained.

Consider **independent claim 1**. The applicant has amended claim 1 to include additional limitations. The examiner has therefore modified the rejection for claim 1 to 103 (a) based on the previously cited reference (Beadle et al.) and a new reference (Anne et al.) that discloses the amended features of claim 1. The applicants argue that selecting the DSL button (or other buttons) does not cause the DSL connection to become modifiable. The examiner does not agree with the argument. After the DSL button is clicked, the user has the option to either accept the default server or click an option to select another server which is less congested. The options to modify a connection are shown in the desktop area marked 503. The applicants are directed in the Beadle et al. reference to column 6, lines 58-67 through column 7, lines 1-30 that further describe these options. The applicants further argue about the "active area", further stating that frame 510 in Fig. 5A is not an active area. The examiner was not able to locate frame 510. However, according to applicants own definition of "empty area" on the desktop is the area that does not include the connection icon, such an area

qualifies as an active area (see paragraph 0075, lines 3-5). Therefore, areas 507, 511, 513 and 515 have been interpreted by the examiner to be active areas. Furthermore, column 11, lines 20-32 disclose that a “switch connection” button may be included in client browser for switching between connection states as shown in Fig. 5B, which corresponds to appearance of a new connection window. The applicants further argue that the cited Fig. 10 and column 10, lines 6-16 do not adequately disclose displaying of a second connection icon within the connection management window. The examiner respectfully disagrees. As shown in Fig. 5A, the satellite icon is shown with dashed lines, indicating presently unavailable connection (column 6, lines 66-67 through column 7, lines 1-2). Later when this icon 505C is clicked to establish a satellite connection, a second connection is established as shown in Fig. 10. Finally, the applicants argue that Beadle et al. do not disclose a first connection between the remote computing device and a first local computing device, and the second connection icon representing a second connection between the remote computing device and a second local computing device. As already explained above with reference to Fig. 10, two icons, satellite and DSL, represent two different connections between a remote computing device (client browser) and a first local computing device (server for DirectPC disclosed in column 10, lines 6-21), and the second connection icon (DSL) represents a second connection between the remote computing device and a second local computing device (server for financial transactions).

Therefore, the examiner has concluded that the cited references provide adequate disclosure and support to teach all the features of **claim 1, which therefore**

remains rejected. Other independent claims 7, 17 and 21 are rejected for the same reasons. All dependent claims 2-6, 8-12, 18-20 and 22-24 are also rejected based on the cited references. The new claims 25-28 are rejected based on the previously cited Beadle et al. reference and the new Anne et al. reference.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any response to this Office Action should be **faxed to (571) 273-8300 or mailed to:**

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Art Unit: 2143

Hand-delivered responses should be brought to

Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Kishin G. Belani whose telephone number is (571) 270-1768. The Examiner can normally be reached on Monday-Thursday from 6:30 am to 5:00 pm.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Nathan Flynn can be reached on (571) 272-1915. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you

have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free) or 703-305-3028.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist/customer service whose telephone number is (571) 272-0800.

Kishin G. Belani

K.G.B./kgb

February 28, 2008

/Kenny S Lin/
Kenny S Lin
Primary Examiner